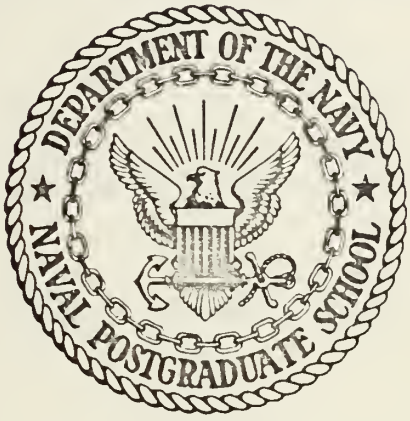
NUMERICAL SOLUTION OF THE FOKKER-PLANCK EQUATION FOR THE FIRST ORDER PHASE-LOCKED LOOP

Allan Rutherford

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

NUMERICAL SOLUTION OF THE
FOKKER-PLANCK EQUATION FOR
THE FIRST ORDER PHASE-LOCKED LOOP

by

Allan Rutherford

Thesis Advisor: J.E. Ohlson

September 1972

Approved for public release; distribution unlimited.

Numerical Solution of the Fokker-Planck Equation for The First Order Phase-Locked Loop

by

Allan Rutherford
Ensign, United States Navy
B.S., California State University, Northridge, 1970

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1972

ABSTRACT

The transient behavior of the phase error probability density function of a first-order phase-locked loop in the presence of additive white Gaussian noise is determined using numerical techniques. In addition, the modulo-2π probability density function and the cycle slipping density function are found. The algorithm used in these analyses involves numerical integration of a normalized and factored Fokker-Planck equation. Results are shown for cases involving various signal-to-noise ratios and initial conditions.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

My thanks to John Ohlson who has made the last nine months the most rewarding period of my entire formal education process.

I. INTRODUCTION

A. BACKGROUND INFORMATION

The space age spurred great interest in the development of phase-locking techniques since fixed tuned receivers were impractical for receiving signals experiencing a wide range of doppler shifts. Due to the inherent nonlinearity of the loop phase comparator, early attempts to analyze phase-locked loop behavior involved the use of graphical phase plane methods which are summarized by Viterbi [1]. The initial use of Fokker-Planck techniques to determine the steady state probability distribution of the first order loop phase error was accomplished by Tikhonov [2],[3]. Since then much work has been done using the Fokker-Planck method of analysis but little has been published dealing with the transient behavior of the phase-error process.

Various techniques have been employed in attempting to statistically describe the phase-error process of first order phase-locked loops. Stationary phase-error distributions are well documented by Viterbi [1] and Tikhonov [2] and [3]. Transient solutions of the Fokker-Planck equation have been obtained by using eigenfunction expansions [4] and by exploiting the similarities between the Fokker-Planck equation and a one-dimensional heat-flow equation [5]. La Frieda [4] arrived at a statistical description of the reduced modulo-2π phase-error process by applying

separation of variables. This application yielded an eigenvalue problem which he solved numerically. Dominiak and Pickholtz [5] arrived at a direct numerical solution to the transient behavior of the phase-locked loop (PLL) by subjecting the Fokker-Planck equation to the same numerical procedures as was provided for the one-dimensional heat-flow equation by von Neumann and Richtmyer [6]. Their resulting partial difference equation was solved using a digital computer.

This study was started in the belief that the previous work in the analysis of the transient behavior of the phase-locked loop could be expanded and improved. La Frieda's technique is applicable only for the reduced modulo-2π solution and does not describe the actual behavior of the phase-error process. Dominiak and Pickholtz create serious doubt as to the completeness and validity of their technique in that their conlcusion of insignificant buildup of probability mass at $\phi = \pm k2\pi$ conflicts greatly with qualitative estimates of Viterbi [1] and Lindsey [7] and with one's intuition. It was also felt that Dominiak and Pickholtz did not make a good choice of SNR for their numerical examples. The stationary probability density function for their examples is so flat that any buildup of probability mass would probably be very small.

In this paper the Fokker-Planck equation is normalized using several constants defined for the purpose [1] and [5]. The resulting equation is then expressed in terms of the

product of two functions, one of which is independent of time. The other function, which is much smoother than the original probability density, is found numerically on a digital computer as a function of time. Multiplying the two yields the solution to the Fokker-Planck equation which quantitatively describes the transient behavior of the phase-error process of a first order phase-locked loop. It is found that this procedure is very efficient computationally and readily provides numerical results describing the phase error process, the modulo-2π phase error process, and the cycle slipping statistics.

B. REVIEW OF PHASE-LOCKED LOOPS

A phase-locked loop is a device which automatically controls an oscillator or periodic function generator so as to operate at a constant phase angle relative to a reference signal source. Figure 1 shows the fundamental hardware structure of the loop with the appropriate signals. The three major components of the PLL are:

a) A multiplier which multiplies the input signal to the loop and the output signal of the voltage controlled oscillator, i.e.,

$$x(t) = 2AK_1 \sin\theta(t)\cos\theta'(t)$$

$$= AK_1\{\sin[\theta(t)-\theta'(t)] + \sin[\theta(t) + \theta'(t)]\}, \quad (1)$$

where $\theta(t)$ and $\theta'(t)$ are the input signal phase and the voltage controlled oscillator output phase respectively.

b) A linear low-pass filter with zero-input response $e_o(t)$ and impulse response $f(t)$ which filters out the sum-frequency component of $x(t)$ while passing a filtered version of the difference-frequency component, i.e.,

$$e(t) = e_o(t) + \int_0^t x(t-\lambda)f(\lambda)d\lambda \quad t \geq 0 \ . \tag{2}$$

c) A voltage controlled oscillator (VCO), the output frequency of which is equal to the sum of its quiescent frequency with the product of its input voltage and a proportionality constant, i.e.,

$$\frac{d\theta'(t)}{dt} = \omega_o + K_2 e(t). \tag{3}$$

Following through on the mathematics [1] it can be shown that for $e_o(t) = 0$,

$$\frac{d\phi'(t)}{dt} = \omega_o + K_2 \int_0^t f(t-\lambda)AK_1 \sin[\theta(\lambda)-\theta'(\lambda)]d\lambda. \tag{4}$$

For phase error $\phi(t) \overset{\Delta}{=} \theta(t)-\theta'(t)$ and total loop gain $K = K_1K_2$

$$\frac{d\phi(t)}{dt} = \frac{d\theta(t)}{dt} - \omega_o - AK\int_0^t f(t-\lambda)\sin\phi(\lambda)d\lambda. \tag{5}$$

Letting

$$\theta_1(t) = \theta(t) - \omega_o t \tag{6}$$

$$\theta_2(t) = \theta'(t) - \omega_o t, \tag{7}$$

then

$$\phi(t) = \theta_1(t) - \theta_2(t) \tag{8}$$

and

$$\frac{d\phi(t)}{dt} = \frac{d\theta_1(t)}{dt} - AK \int_0^t f(t-\lambda)\sin\phi(\lambda)d\lambda \tag{9}$$

which implies the PLL model shown in Fig. 2.

Additive noise at the input which is narrowband Gaussian noise n(t) with zero mean and spectral density $\frac{N_o}{2}$ near the signal frequency, can be expressed as

$$n(t) = \sqrt{2}\{n_1(t)\sin\omega_o t + n_2(t)\cos\omega_o t\} \tag{10}$$

where $n_1(t)$ and $n_2(t)$ are independent, zero mean Gaussian processes of identical spectral densities, which are the same as the spectral density of n(t) but translated downward in frequency so as to be centered about zero frequency. A derivation similar to the derivation of (9) is made [1] with the noise having the form

$$n'(t) = -n_1(t)\sin\theta_2(t) + n_2(t)\cos\theta_2(t) \tag{11}$$

where n'(t) may be treated as a white Gaussian random process with spectral density $\frac{N_o}{2}$. This leads to the model

of the PLL in Figure 3, [1]. The equation of operation of the phase-locked loop is now given by

$$\frac{d\phi(t)}{dt} = \frac{d\theta_1(t)}{dt} - K\int_0^t f(t-\lambda)[A\sin\phi(\lambda) + n'(\lambda)]d\lambda. \tag{12}$$

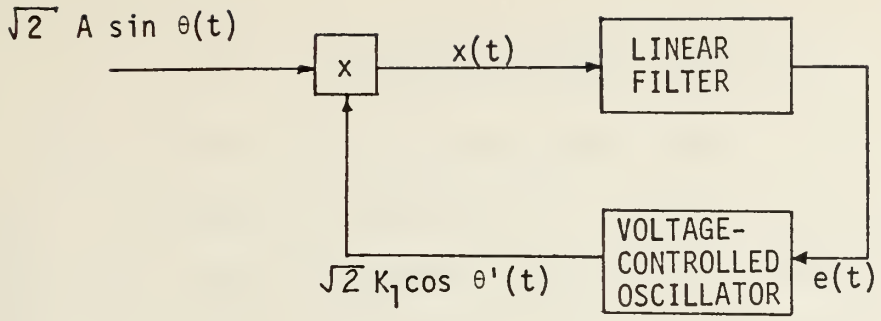


FIGURE 1. Fundamental Hardware Structure of the Phase-Locked Loop.

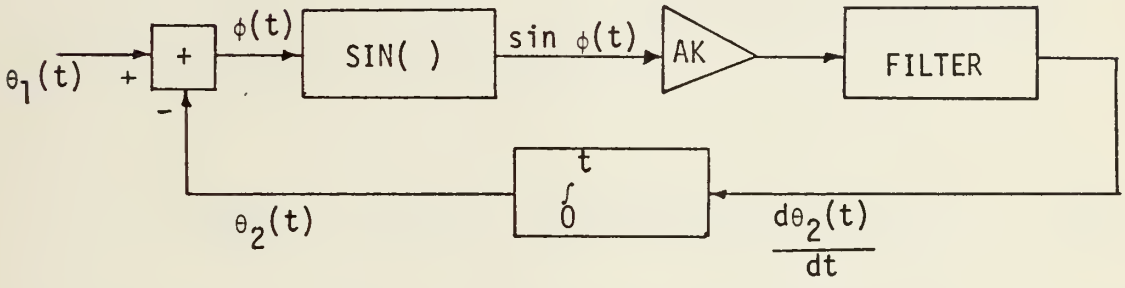


FIGURE 2. Phase-Locked Loop Modeled Without Noise.

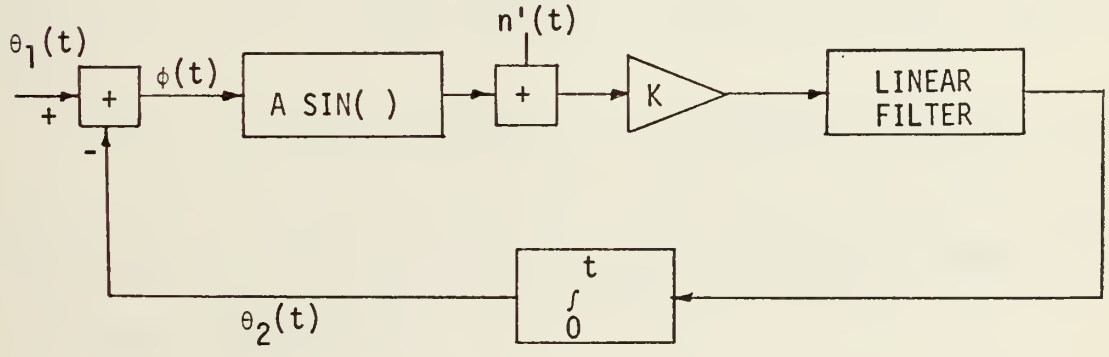


FIGURE 3. Phase-Locked Loop Modeled With Additive Noise.

II. FOKKER-PLANCK EQUATION

Let a first-order phase-locked loop have an input signal $e_i(t)$ of constant frequency ω, i.e., $\theta(t)=\omega t$, and a voltage controlled oscillator with quiescent frequency ω_o. Additive thermal noise, $n(t)$, at the front end of a receiver is a zero-mean Gaussian process whose spectral density is essentially flat over the frequency range of the receiver and may be assumed to be white with value $\frac{N_o}{2}$. The equation describing the loop operation [1] is from (12)

$$\frac{d\phi(t)}{dt} = (\omega-\omega_o) - K[A\sin\phi(t) + n'(t)] \qquad (13)$$

where $\phi(t)$ is the difference between the incoming signal phase and the VCO output phase and K is the total loop gain.

It can be shown[1] that $\phi(t)$ is a continuous Markov process. The noise $n'(t)$ is a stationary Gaussian process with zero mean and it is nearly white over the frequency range of interest. Therefore,

$$E[n'(t_1)n'(t_2)] = \frac{N_o}{2}\,\delta(t_2-t_1). \qquad (15)$$

Under these conditions, the probability density function $p(\phi,t)$ of the Markov process phase error $\phi(t)$ must satisfy the Fokker-Planck equation [1], i.e.,

$$\frac{\partial p(\phi,t)}{\partial t} = -\frac{\partial}{\partial \phi}\left[(\omega-\omega_o-AK\sin\phi)p(\phi,t)\right] + \frac{K^2N_o}{4}\frac{\partial^2 p(\phi,t)}{\partial \phi^2} \tag{16}$$

with initial condition

$$p(\phi,0) = \delta(\phi-\phi_o) \tag{17}$$

and normalizing condition

$$\int_{-\infty}^{\infty} p(\phi,t)d\phi = 1 \ . \tag{18}$$

It is convenient to express (16) in normalized form. Let

$$B_L = AK/4 \qquad \text{loop-noise bandwidth} \tag{19}$$

$$\alpha = A^2/N_oB_L \qquad \text{signal to noise ratio} \tag{20}$$

$$\gamma = (\omega-\omega_o)/4B_L \qquad \text{frequency ratio} \tag{21}$$

$$\tau = 4B_Lt \qquad \text{normalized time.} \tag{22}$$

Substituting (19) - (22) into (16) yields a normalized Fokker-Planck equation which is

$$\frac{\partial p(\phi,\tau)}{\partial \tau} = \frac{\partial}{\partial \phi}\left[(\sin\phi-\gamma)p(\phi,\tau)\right] + \frac{1}{\alpha}\frac{\partial^2 p(\phi,\tau)}{\partial \phi^2} \tag{23}$$

An analytic solution to (23) has never been found.

It is apparent that by considering only very small phase errors, i.e., $\sin\phi \approx \phi$, then (13) can be linearized. Making

the same approximation in (23) gives a closed form solution which satisfies the initial and boundary conditions on $p(\phi,\tau)$. This solution is

$$p(\phi,\tau) = \left[\frac{\alpha}{2\pi(1-e^{-2\tau})}\right]^{1/2} \exp\left[\frac{\alpha[\phi-\gamma-(\phi_o-\gamma)e^{-\tau}]^2}{2(1-e^{-2\tau})}\right] \quad (24)$$

which is Gaussian with

$$\text{mean} = m(\tau) = \gamma + (\phi_o - \gamma)\, e^{-\tau} \quad (25)$$

and

$$\text{variance} = \sigma^2(\tau) = (1 - e^{-2\tau})/\alpha \ . \quad (26)$$

It is the Gaussian nature of (24) which provides the information necessary to choose the initial boundary condition function for the numerical algorithm used in the solution of (23). However, before the selection of an initial condition is discussed, it is appropriate to analyze the technique utilized in arriving at the numerical algorithm used in this study.

III. TRANSIENT ANALYSIS

A. FUNDAMENTAL TECHNIQUE

The fundamental technique employed in this study is the treatment of the normalized solution to the Fokker-Planck loop equation as the product of two functions, i.e.,

$$p(\phi,\tau) = g(\phi,\tau)\, \exp[\alpha(\cos\phi + \gamma\phi)] \,. \qquad (27)$$

This exponential factor was selected on both mathematical and intuitive grounds. Intuitively, one would expect that after having slipped a cycle the loop dynamics would again exhibit some measure of steady state behavior prior to another slip. Lindsey [7] makes a similar observation in his analysis of cycle-slipping probabilities. At any one time the loop phase-error tends to wobble about one of an infinite set of points spaced by 2π radians. It seems reasonable that the phase error probability density should resemble a mod-2π stationary density function in each 2π interval with a broad taper away from the origin reflecting the random walk, i.e. cycle slipping, nature of the problem.

The "envelope" encompassing this succession of quasi-mod-2π functions can be easily imagined to be much broader and smoother than the $p(\phi,\tau)$ function. It seems plausible that the function $g(\phi,\tau)$ from (27) is this smooth "envelope". The desirability of this observation is that the "smoothness" of $g(\phi,\tau)$ would allow a much coarser grid to be used in a

numerical integration scheme than would be possible with the function $p(\phi,\tau)$.

Applying (27) to the normalized F-P equation (23) one obtains

$$\frac{\partial g(\phi,\tau)}{\partial \tau} = (\gamma-\sin\phi)\,\frac{\partial g(\phi,\tau)}{\partial \phi} + \frac{1}{\alpha}\,\frac{\partial^2 g(\phi,\tau)}{\partial \phi^2} \,. \qquad (28)$$

It is now apparent that solving for $g(\phi,\tau)$ is computationally simpler than solving for $p(\phi,\tau)$. And, since the exponential factor is a constant for each value of ϕ along the ϕ-axis, $p(\phi,\tau)$ can be reconstructed readily using (27).

Equation (28) is evaluated numerically by using the central-difference method to approximate the derivatives, such that with $\Delta\phi$ and $\Delta\tau$ being the increments in ϕ and τ respectively, then

$$\frac{\partial g(\phi,\tau)}{\partial \tau} \simeq \frac{g(\phi,\tau+\Delta\tau)-g(\phi,\tau)}{\Delta\tau} \qquad (29)$$

$$\frac{\partial g(\phi,\tau)}{\partial \phi} \simeq \frac{g(\phi+\Delta\phi,\tau)-g(\phi,\tau)}{\Delta\phi} \qquad (30)$$

$$\frac{\partial^2 g(\phi,\tau)}{\partial \phi^2} \simeq \left[\frac{g(\phi+\Delta\phi,\tau)-g(\phi,\tau)}{\Delta\phi} - \frac{g(\phi,\tau)-g(\phi-\Delta\phi,\tau)}{\Delta\phi}\right] \frac{1}{\Delta\phi}$$

$$\simeq \frac{g(\phi+\Delta\phi,\tau) - 2g(\phi,\tau) + g(\phi-\Delta\phi,\tau)}{(\Delta\phi)^2} \qquad (31)$$

and it follows directly that

$$g(\phi,\tau+\Delta\tau)=g(\phi,\tau)+(\gamma-\sin\phi)\frac{\Delta\tau}{\Delta\phi}[g(\phi+\Delta\phi,\tau)-g(\phi-\Delta\phi,\tau)] \qquad (32)$$

$$+ \frac{1}{\alpha}\,\frac{\Delta\tau}{(\Delta\phi)^2}\,[g(\phi+\Delta\phi,\tau)-2g(\phi,\tau)+g(\phi-\Delta\phi,\tau)].$$

B. OTHER CONSIDERATIONS

1. Initial Condition

It is the response of (23) to an input impulse function (17) which is sought. However, it was felt that a finite difference method algorithm acting on the extreme discontinuities of a numerical delta function of $\Delta\phi$ width and $1/\Delta\phi$ height might result in significant errors in the results. Therefore the Gaussian nature of (24) was utilized in choosing an initial boundary condition for the algorithm. The boundary condition selected was a Gaussian function which corresponds very nearly to the response of (23) at some very small time, assuming the impulse function (17) existed at time $\tau = 0$. To check the accuracy of this boundary condition three techniques were employed. The first was to plot the variance of the calculated phase error versus time to determine whether or not the loop is operating in its linear region; if so, the variance vs. time curve would extrapolate linearly back to an equivalent origin, where the variance is zero at $\tau = 0$. The second was to actually test the phase error response to the numerical delta function mentioned above. The third was to compare stationary and transient mod-2π results with those already published [1], [4] and [5].

When the initial condition is the Gaussian function, the equivalent time, τ_o, at which this condition would occur if $p(\phi,0) = \delta(\phi-\phi_o)$ is given by (26) as

$$\sigma_o^2 = \sigma^2(\tau_o) = \frac{1 - e^{-2\tau_o}}{\alpha} \tag{33}$$

and it follows that

$$\tau_o = -\frac{1}{2}[\ln(1-\alpha\sigma_o^2)]. \tag{34}$$

In this study $B_L t = 0$ and $B_L t = \frac{\tau_o}{4}$ are the times at which the equivalent delta function and the Gaussian initial condition existed respectively. In [5], however, $B_L t = 0$ was taken to be the time of existence of their Gaussian initial condition, and this does not truly represent the dependency of $p(\phi,\tau)$ on time.

2. End-value Algorithm

The use of a digital computer in evaluating this numerical integration scheme requires a finite interval which comprises the sample points along the ϕ-axis. A potentially large source of error in any computations performed on this fixed interval is the method by which the end points of the interval are evaluated after each $\Delta\tau$ step. The central-difference method algorithm in this study, if applied exclusively, would result in an error propagating inward one $\Delta\phi$ increment per $\Delta\tau$ increment starting with the first $\Delta\tau$ increment. This error would make the procedure useless for large τ unless the total ϕ-interval were made prohibitively large. To minimize error from this source,

a sixth-order Stirling extrapolation formula is used to evaluate both of the end points.

3. Stability

Stability in a numerical algorithm implies that the errors generated at each step of the numerical integration process tend to decay as the solution progresses in time. The method used in this study to determine the stability conditions is patterned directly after von Neumann's Fourier series technique [6] using the format presented in [8]. The mesh is assumed sufficiently small such that the coefficients in (32) may be treated as constants over a region which is small but contains many mesh points. For N sample points spaced $\Delta\phi$ radians along the ϕ-interval, the initial errors, i.e. $\tau = 0$, which are assumed to be located at the sample points can be expressed for each point in complex exponential form as

$$E_k = \sum_{n=0}^{N-1} A_n e^{j\beta_n k(\Delta\phi)} , \quad k = 0,1,2,\ldots,N-1 \qquad (35)$$

where

$$\beta_n = \frac{n2\pi}{(N-1)(\Delta\phi)} . \qquad (36)$$

Because the finite-difference equations will always be linear and, therefore, separate solutions additive, one need only consider the propagation of the error due to a single term, e.g. $e^{j\beta k(\Delta\phi)}$. The coefficient A_n is a constant and

can be neglected; and for simplicity the subscript n on β is dropped.

To investigate the propagation of this error for increasing τ, one must find a solution of the finite-difference equation which reduces to $e^{j\beta k(\Delta\phi)}$ when $\tau = \ell(\Delta\tau) = 0$. $\Gamma_{k\ell}$ is defined as the error at the k^{th} point along the ϕ interval at the ℓ^{th} iteration along τ. Assume that the error has the form

$$\Gamma_{k\ell} = e^{j\beta k(\Delta\phi)}\, e^{\lambda\tau} = e^{j\beta k(\Delta\phi)}\, e^{\lambda\ell(\Delta\tau)}$$

$$= e^{j\beta k(\Delta\phi)}\, \xi^{\ell} \qquad (37)$$

where $\xi = e^{\lambda(\Delta\tau)}$ and λ, in general, is a complex constant. Clearly, (37) reduces to $e^{j\beta k(\Delta\phi)}$ when $\ell = 0$. Also, the error will not increase for increasing τ provided

$$|\xi| \leq 1. \qquad (38)$$

Since the error $\Gamma_{k\ell}$ satisfies the same difference equations as $g(\phi,\tau)$, substitution of (37) into (32) and subsequent division by $e^{j\beta k(\Delta\phi)}\, \xi^{\ell}$ yields

$$\xi = 1 + (\gamma - \sin\phi)\frac{\Delta\tau}{\Delta\phi}\{\exp[j\beta(\Delta\phi)] - \exp[-j\beta(\Delta\phi)]\}$$

$$+ \frac{1}{\alpha}\frac{\Delta\tau}{(\Delta\phi)^2}\{\exp[j\beta(\Delta\phi)] - 2 + \exp[-j\beta(\Delta\phi)]\}$$

$$= 1 + 2(\gamma - \sin\phi)\frac{\Delta\tau}{\Delta\phi}\sin[\beta(\Delta\phi)] + \left(\frac{2}{\alpha}\right)\frac{\Delta\tau}{(\Delta\phi)^2}\{\cos[\beta(\Delta\phi)] - 1\}. \qquad (39)$$

If one assumes that the $2(\gamma-\sin\phi)\frac{\Delta\tau}{\Delta\phi}\sin[\beta(\Delta\phi)]$ term is negligible for large N and very small $\Delta\phi$, then from (38) the following inequality is established:

$$\Delta\tau < \frac{\alpha}{2}(\Delta\phi)^2. \tag{40}$$

For numerical calculations let $\Delta\tau$ and $\Delta\phi$ be related as

$$\Delta\tau = \frac{\alpha}{4}(\Delta\phi)^2. \tag{41}$$

For values of α in the vicinity of unity and for $\Delta\phi << 1$, then $\Delta\tau << \Delta\phi$. Hence, the dropping of the term containing $\frac{\Delta\tau}{\Delta\phi}$ is justified.

4. Cycle Slipping

The inclusion of absorbing boundaries around some sub-interval of ϕ, e.g. $[-\pi,\pi]$, enables one to determine the probability that the phase error has not exceeded the limits of that sub-interval at any point in time. As the density function spreads out, any probability mass which reaches the limits of the sub-interval is removed from the system entirely by forcing the density function to be zero outside the sub-interval. The effect of this technique is to make the remaining area of the density function equal to the probability that the phase error has not reached the limits of that sub-interval.

The cycle-slipping density function is denoted $q(\phi,\tau)$ to distinguish it from the corresponding function

$p(\phi,\tau)$ for the unbounded case. As long as the phase error of the loop remains within the limits of a specified subinterval, the density function $q(\phi,\tau)$, with initial condition $q(\phi,0) = \delta(\phi)$, will satisfy the Fokker-Planck equation (23), i.e.

$$\frac{\partial q(\phi,\tau)}{\partial \tau} = \frac{\partial}{\partial \phi}[(\sin\phi - \gamma)q(\phi,\tau)] + \frac{1}{\alpha}\frac{\partial^2 q(\phi,\tau)}{\partial \phi^2}. \qquad (42)$$

Following the development from Viterbi [1], for $\gamma = \phi_o = 0$ the probability that the phase error ϕ has not reached some value ϕ_ℓ by time τ is given by

$$\psi(\tau) = \int_{-\phi_\ell}^{\phi_\ell} q(\phi,\tau)d\phi \qquad (43)$$

where

$$q(\phi_\ell,\tau) = q(-\phi_\ell,\tau) = 0 \quad \text{for all } \tau . \qquad (44)$$

The fundamental difference between $q(\phi,\tau)$ and $p(\phi,\tau)$ is that $q(\phi,\tau)$ is not strictly a probability density function, unless it is normalized by $\psi(\tau)$. The numerical algorithm used to generate $q(\phi,\tau)$ is the same as for generating $p(\phi,\tau)$ except that the end points $q(\phi_\ell,\tau)$ and $q(-\phi_\ell,\tau)$ are forced to be zero. The end points for $p(\phi,\tau)$ are found by extrapolation, thus simulating an infinite interval.

In this study an analysis was made of the cycle slipping probabilities for the primary cycle slip, i.e.,

$$[-\phi_\ell,\phi_\ell] = [-\pi,\pi] \qquad (45)$$

and for the primary half-cycle slip, i.e.,

$$[-\phi_\ell, \phi_\ell] = [-\pi/2, \pi/2] \ . \tag{46}$$

Plots of $\Psi(\tau)$ versus τ show the time dependent behavior of this cycle slipping probability.

IV. RESULTS

The entire numerical analysis of this study was programmed for an IBM-360/67 computer using Fortran IV language, level G. Double precision computation was used exclusively to minimize round-off and truncation errors. The initial condition on $p(\phi,\tau)$ in (23) was a Gaussian function with mean ϕ_o and standard deviation $\sigma_o = \pi/20$. The ϕ-interval of investigation was $[-11\pi,11\pi]$ with $g(\phi,\tau)$ and, therefore, $p(\phi,\tau)$ being computed over the entire interval to allow for the non-symmetrical conditions $\gamma \neq 0$ and $\phi_o \neq 0$. Stability considerations specify that $\Delta\tau < (\alpha/2)(\Delta\phi)^2$, and to accommodate this requirement $\Delta\tau$ was chosen such that $\Delta\tau = (\alpha/4)(\Delta\phi)^2$. The sampling interval along the ϕ-axis was selected to be $\Delta\phi = \pi/50$ except in the cycle slipping calculation where $\Delta\phi = \pi/100$. For the examples which will be shown here, it was found that negligible improvements were obtained for smaller values of $\Delta\phi$ and σ_o. In fact, a "delta function" of width $\Delta\phi$ and height $1/\Delta\phi$ was used as the initial condition on $p(\phi,\tau)$ in several cases, and the results obtained varied only by approximately 1 part in 20,000 from those obtained with the initial Gaussian condition above. Table I indicates the various conditions which were studied. The specific values were chosen so as to provide comparison with previously published work [1], [4], [5]. Also included in the table are the accuracy

considerations for this study expressed as areas of the probability density function after the indicated number of time steps.

TABLE I

α	ϕ_o	γ	$\Delta\tau$	Number of Time Steps	Area Under Curves
0.1	0	0	$\pi^2 x 10^{-5}$	20,000	1.000
1.0	0	0	$\pi^2 x 10^{-4}$	40,000	1.000
1.0	0	$\sin(\pi/4)$	$\pi^2 x 10^{-4}$	20,000	1.000
2.0	$\pi/2$	0	$2\pi^2 x 10^{-4}$	20,000	1.000

Figures 4, 5, 6, and 7 are graphs relating to the conditions $\alpha = 1.0$ and $\gamma = \phi_o = 0$. Figure 4 is a sample of the dynamics of the $g(\phi,\tau)$ function from (27). It demonstrates the "smoothness" of the function with respect to the $p(\phi,\tau)$ function which is shown in Figure 5. Transitional and stationary modulo-2π densities are depicted in Figure 6. The stationary modulo-2π density function is seen to be a reproduction of the curve determined by Viterbi [1]. Of particular interest from Figures 5 and 6 is that the $p(\phi,\tau)$ function continues to spread out with buildups of probability mass at the $\pm k2\pi$ points even though the steady state modulo-2π density function has been reached. Figure 7 is a plot of the variance of the phase error probability density function versus time. It is interesting to note the linear nature of the curve for $B_L t > 1.2$.

La Frieda's eigenfunction analysis [4] resulted in transitional and steady state mod-2π densities which were reproduced using the algorithm of this study. Figure 8 is included to show the $p(\phi,\tau)$ curves corresponding to each of the times used in [4] with an additional curve to demonstrate clearly the asymmetric probability mass buildup associated with the conditions $\alpha = 2$, $\gamma = 0$, and $\phi_o = \pi/2$.

Inasmuch as La Frieda used a different normalizing scheme, $D\tau$, in his study, equivalent times, i.e. $B_L t = D\tau/2$, are shown on the graph for ease of reference. The asymmetry of probability mass distribution is due to the main body of $p(\phi,\tau)$ being offset from $\phi = 0$ for some finite period of time while it is shifting. Since $p(\phi,\tau)$ is settling even while it is shifting, more mass will accumulate at $\phi = 2\pi$ than at $\phi = -2\pi$. Once the main body of $p(\phi,\tau)$ is settled about $\phi = 0$, however, the slippage of mass in either direction becomes equally likely and the level of asymmetry will remain constant for increasing time.

Viterbi dealt with a detuned loop which was also analyzed in this study. For this case $\alpha = 1.0$, $\gamma = \sin(\pi/4)$, and $\phi_o = 0$ [1]. Figures 9 and 10 show the mod-2π probability density function and $p(\phi,\tau)$ for these conditions respectively. As before the steady state mod-2π density function determined by this study is a reproduction of that found by Viterbi. As would be expected and is demonstrated

in these two curves, the detuned loop exhibits a definite tendency to slip in the direction of the steady-state phase shift caused by detuning.

Figure 11 is the $p(\phi,\tau)$ function corresponding to $\alpha = 0.1$ and $\gamma = \phi_o = 0$. This set of conditions is one of those selected by Dominiak and Pickholtz [5] in their analysis of the first order PLL by a numerical technique. This curve shows the buildup of probability mass which went undetected in the other study. It is felt that Dominiak and Pickholtz did not find this buildup for several reasons. Firstly, they did not look look enough in their time frame. Secondly, their attempt to deal directly with the $p(\phi,\tau)$ function rather than an intermediate function, e.g. $g(\phi,\tau)$ in this study, resulted in a grid mesh size which, although mathematically correct, was too coarse to detect the subtle and yet significant mass slippage of the probability density function. Finally, from [1] it is apparent that for $\alpha = 0.1$ the mod-2π density function is very flat and is not a good choice of SNR with which to search for probability mass buildup.

Figures 12 and 13 show the dynamics of the cycle slipping density functions, $q(\phi,\tau)$, for primary cycle slips, $[-\pi,\pi]$, and primary half-cycle slips, $[-\pi/2,\pi/2]$. These curves are plotted for the conditions $\alpha = 0.5$ and $\phi_o = \gamma = 0$. In Figures 14 and 15 are the curves which demonstrate how the cycle slipping probability, $\psi(\tau)$, varies with time for

the cases shown in Figures 13 and 14. The function $q(\phi,\tau)$ for the primary cycle slip is essentially identical to the $p(\phi,\tau)$ function until a significant amount of probability mass is lost at the absorbing boundaries at $\phi = \pm\pi$. Until this loss occurs the area under the $q(\phi,\tau)$ curve, i.e. $\psi(t)$, is 1.0. The time at which $\psi(t)$ drops below some value, e.g. 0.5, can be determined from Figure 14. From Figures 13 and 15 similar data may be obtained for the primary half-cycle slip which has absorbing boundaries at $\phi = \pm \frac{\pi}{2}$.

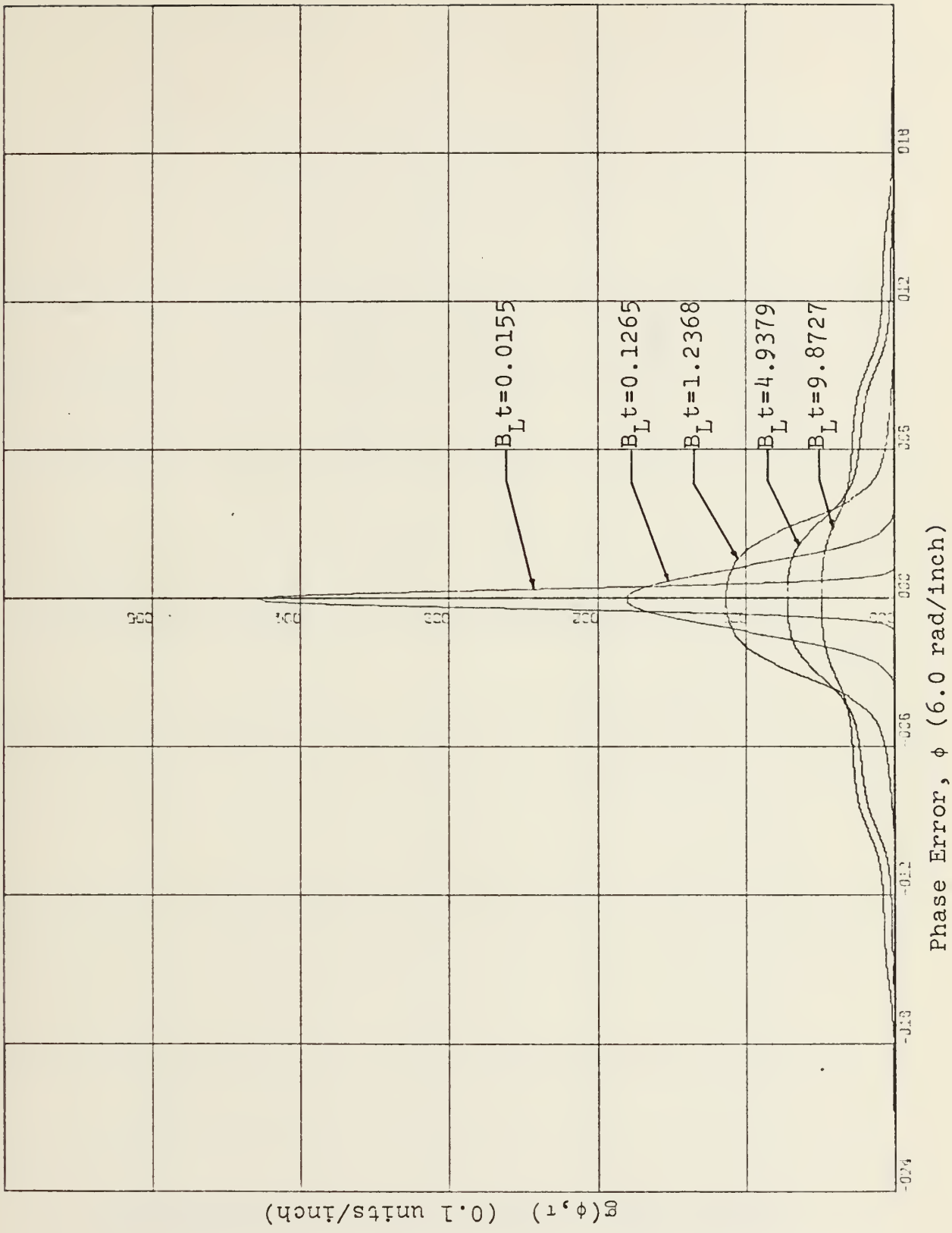


FIGURE 4. $g(\phi,\tau)$ Density Function for α=1.0,$\gamma=\phi_o$=0.

FIGURE 5. Probability Density Function of Phase Error, $\alpha=1.0$, $\gamma=\phi_o=0$.

FIGURE 6. Modulo-2π Probability Density Function of Phase Error, $\alpha=1.0, \gamma=\phi_o=0$.

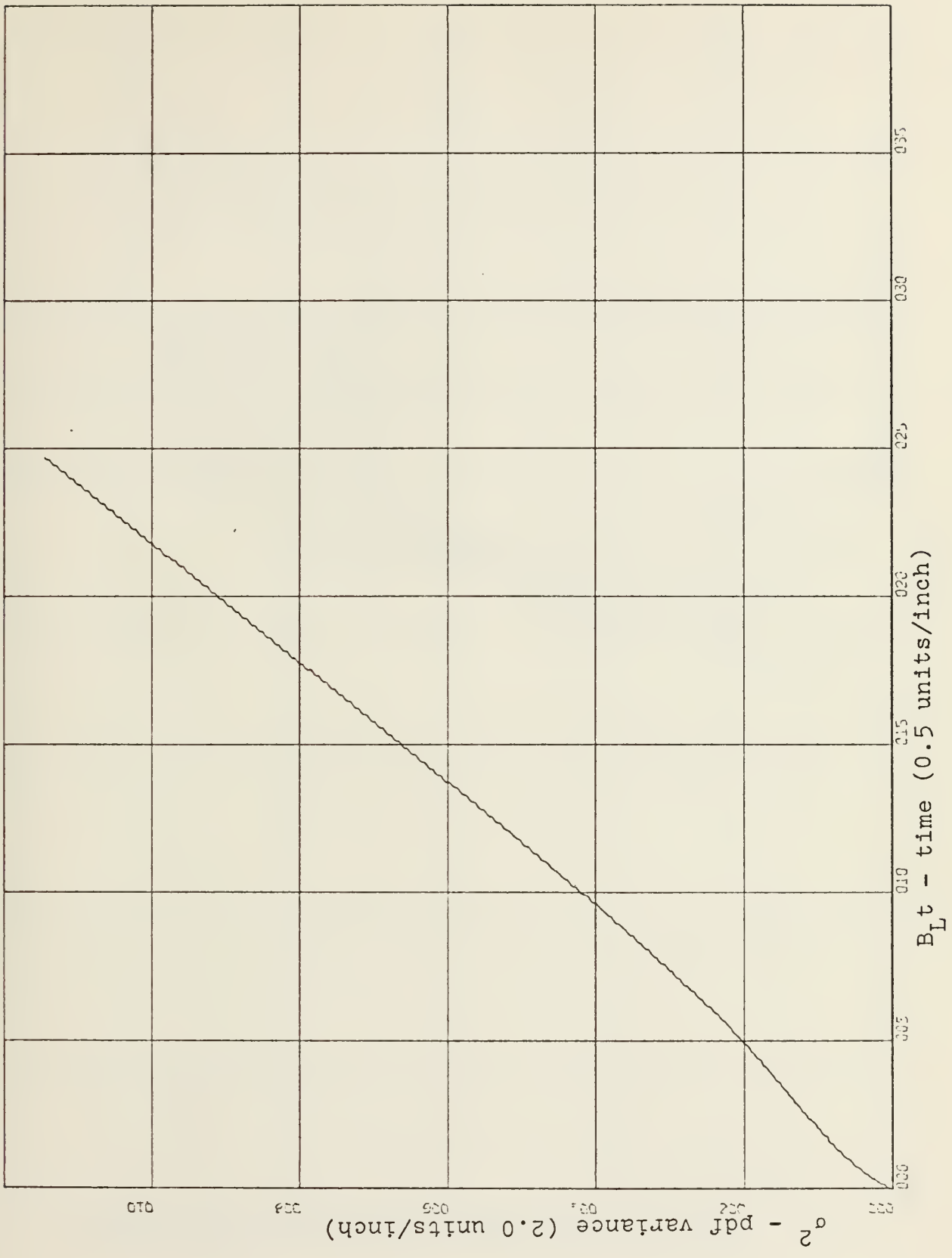


FIGURE 7. Probability Density Function Variance as a Function of Time, α=1.0.

FIGURE 8. Probability Density Function of Phase Error, $\alpha=2.0, \gamma=0, \phi_o=\pi/2$.

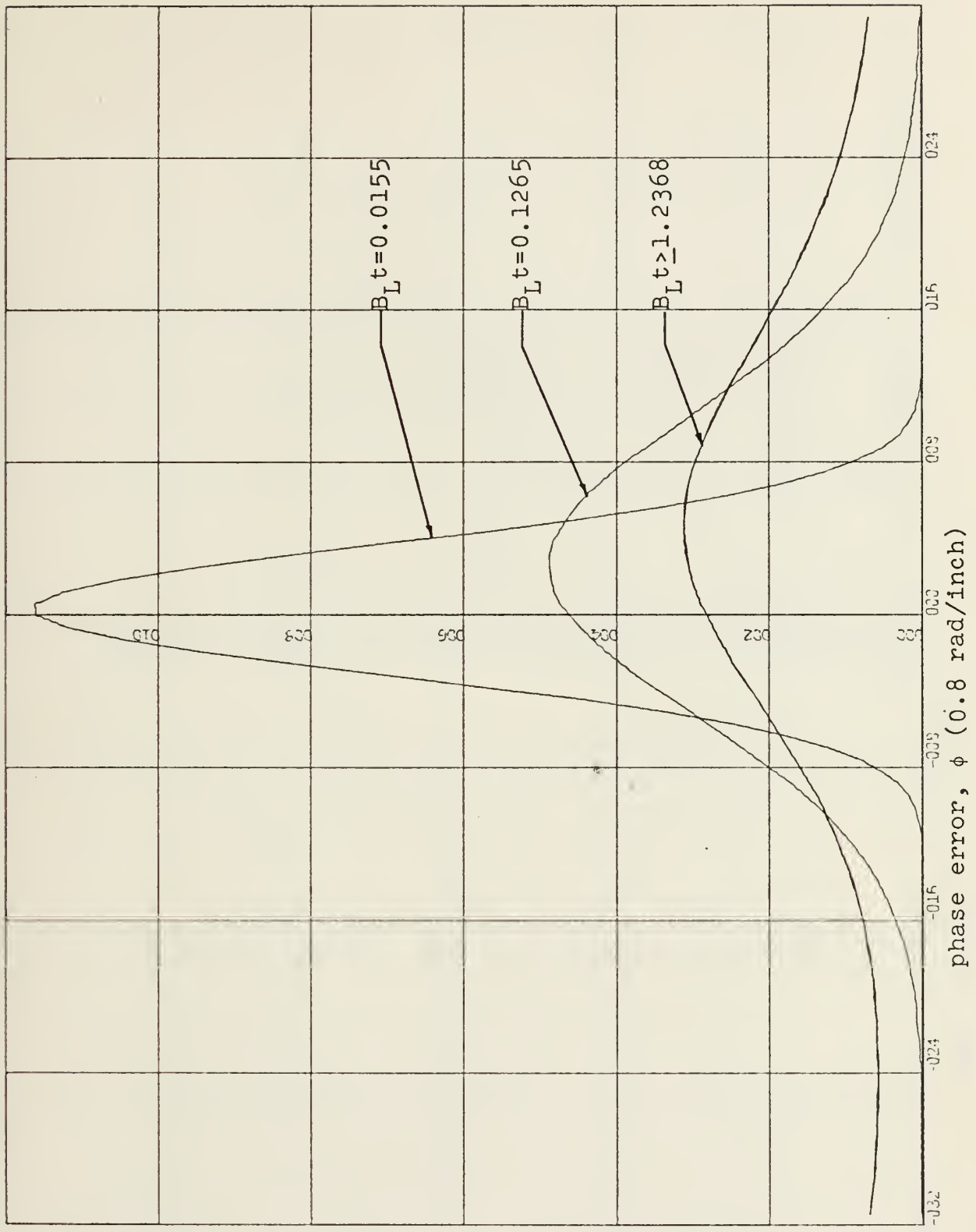


FIGURE 9. Modulo-2π Probability Density Function of Phase Error, $\alpha=1.0$, $\gamma=\sin(\pi/4)$, $\phi_o=0$.

FIGURE 10. Probability Density Function of Phase Error
$\alpha=1.0$, $\gamma=\sin(\pi/4)$, $\phi_o=0$.

FIGURE 11. Probability Density Function of Phase Error α=0.1, $\gamma=\phi_o$=0.

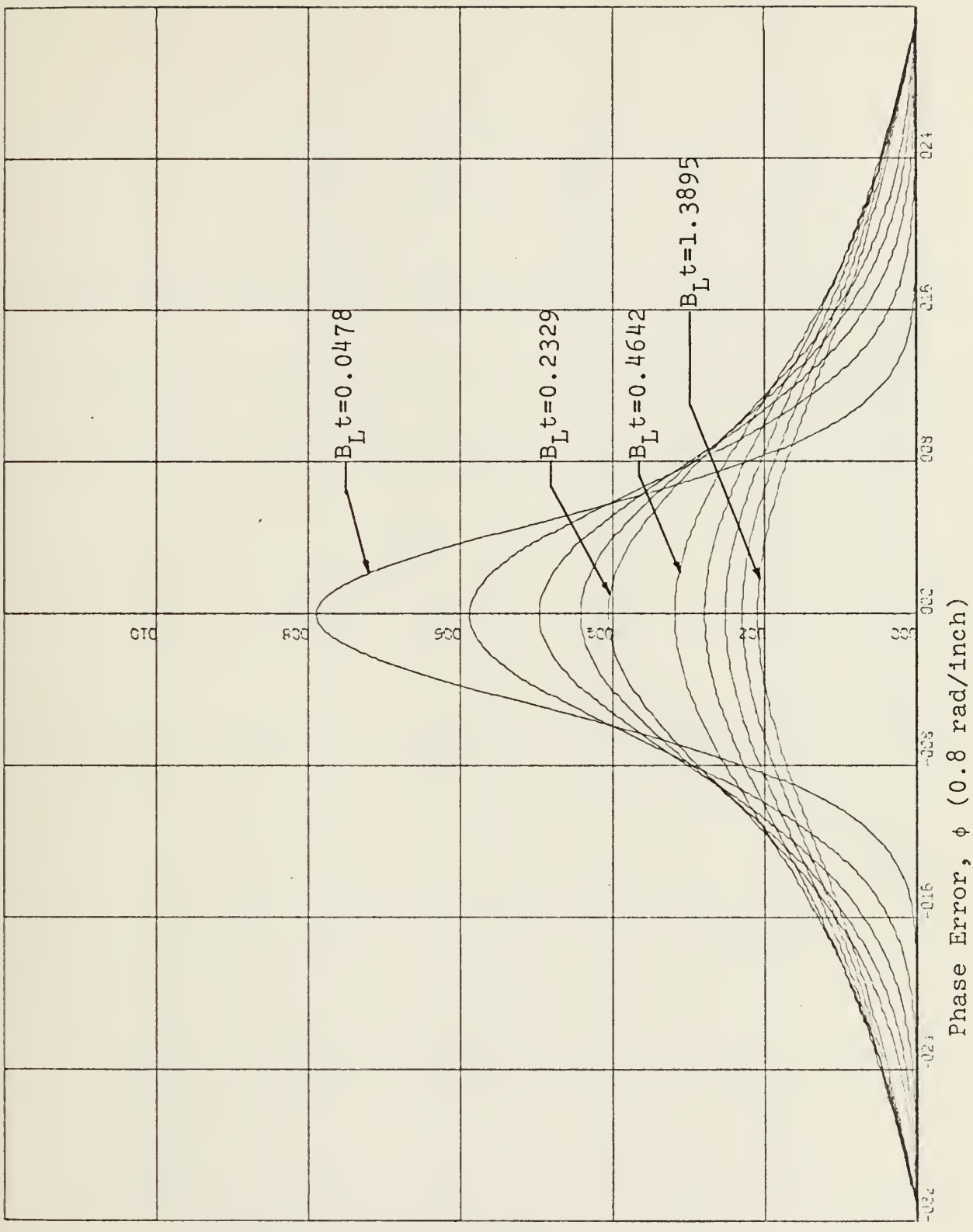


FIGURE 12. Primary Cycle Slip Density Function, α=0.5.

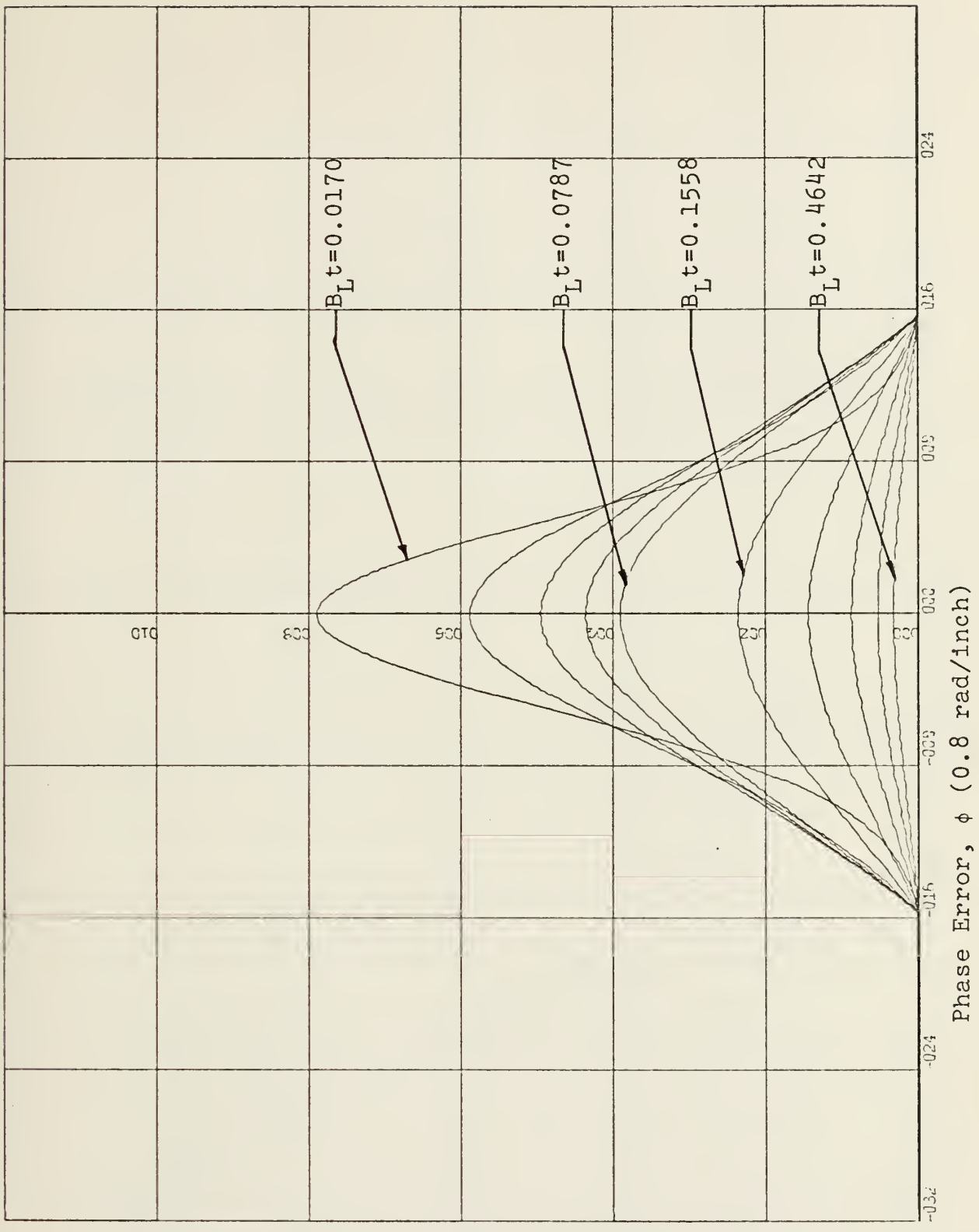


FIGURE 13. Primary Half-Cycle Slip Density Function, α=0.5.

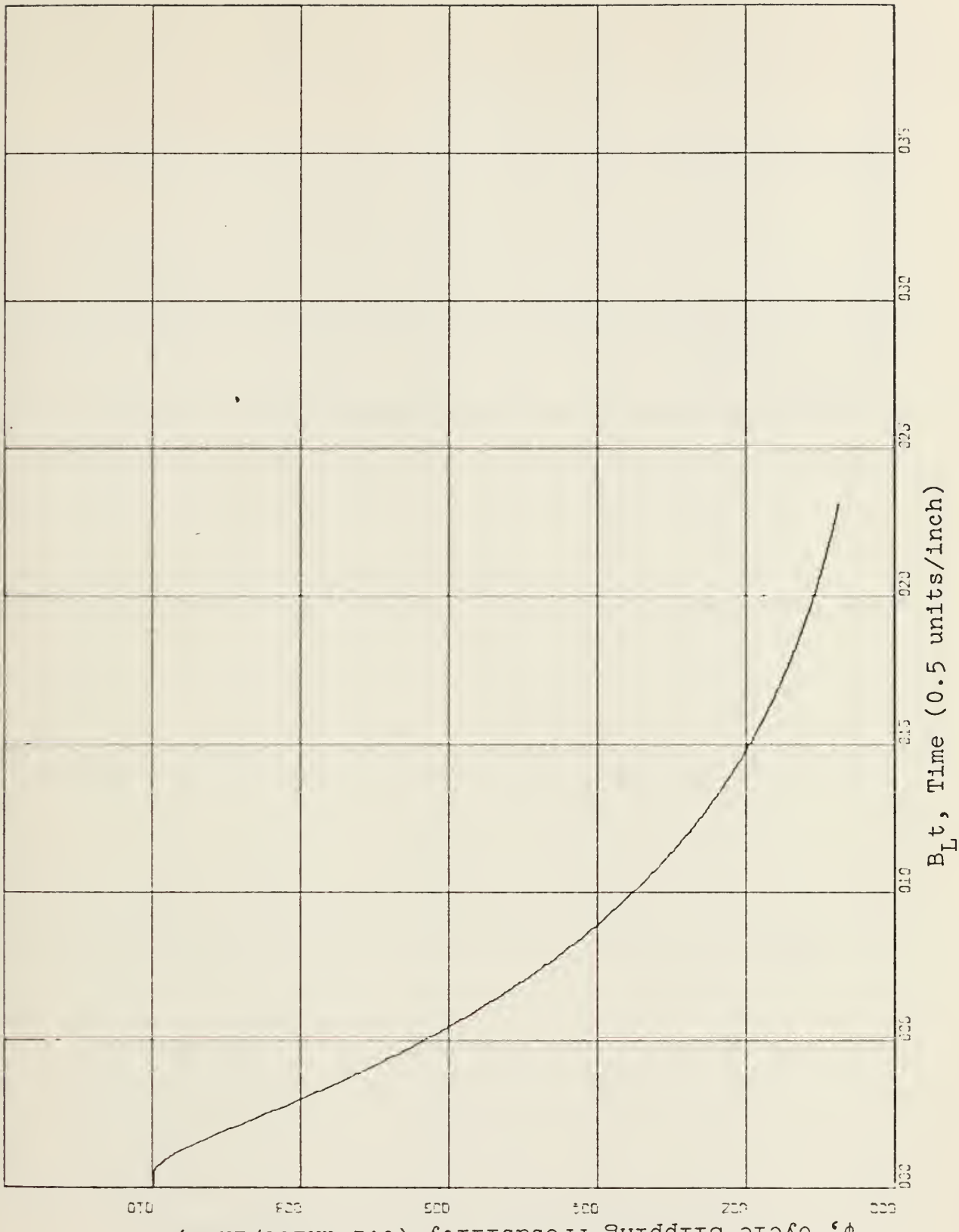


FIGURE 14. Primary Cycle Slipping Probability as a Function of Time, α=0.5.

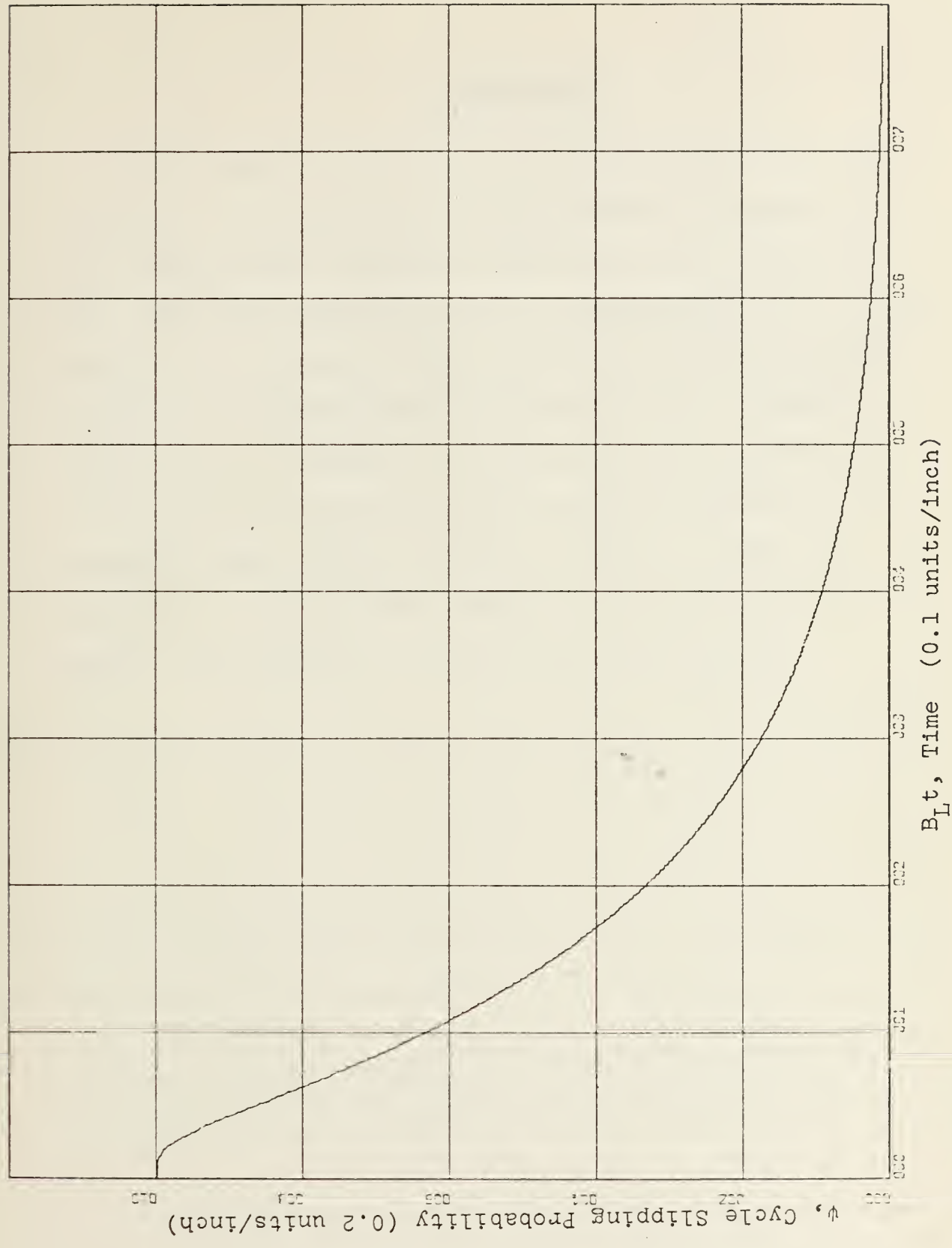


FIGURE 15. Primary Half-Cycle Slipping Probability as a Function of Time, α=0.5.

V. CONCLUSIONS

This study has resulted in the development of a valid technique for numerically finding transient solutions for the Fokker-Planck equation as applied to the first-order PLL. That the technique works is demonstrated by its reproduction of existing published stationary mod-2π solutions in [1] and transient mod-2π solutions in [4]. Results not previously available for the phase-error probability density function and the cycle slipping probability are presented here. Most important, however, is that the qualitative notion of the buildup of probability mass around points spaced 2π radians has been verified.

```
      IMPLICIT REAL*8(A-D,F-H,O-W,Z,$)
      REAL*8 TITLE(12)
      DIMENSION GT(1104), GDT(1104), SINP(1104), COSP(1104), X(1104),
     $Y(1104), ANS(1104), PLOG(1104), DMOD(1104), PHIEQ(1104),
     $PSUM(25), VARX(25), VARY(25)
      REAL LABEL/'    '/
      CALL ERRSET(208, 256, -1, 1, 0)
C
C     SIGNAL TO NOISE RATIO---ALPHA
C     FREQUENCY RATIO---SIN(PI* GAMMA )
C     INTERVAL OF INVESTIGATION---(-A(PI),+A(PI))
C     DELTA PHI STEP SIZE, DPHI=(PI)/B
C     INITIAL CONDITION ON STD DEVIATION, SIGMA=(PI)/C
C     PHASE ANGLE OF INPUT DELTA FCN---PI* PHIZRO
C     NUMBER ITERATIONS ALONG TAU, NUM
C     NNN=1/0 INDICATES NEW DATA CARD  IS/ IS NOT  PRESENT
C     NUMBER COMPUTATIONS ALONG PHI AXIS, N=2*A*B+1
    5 READ(5,6) ALPHA, GAMMA, A, B, C, PHIZRO, NUM, NNN
    6 FORMAT(6F10.5, 2I5)
      N=2*IDINT(A)*IDINT(B)+1
      WRITE(6,13) N
   13 FORMAT('0', 'N=', 2X, I5)
      WRITE(6,11) ALPHA, GAMMA, A, B, C, PHIZRO, NUM, NNN
   11 FORMAT('0', 'INITIAL CONDITIONS', 5X, 6F10.5, 2I5)
C
C     CLEAR ARRAYS
      JT=N+3
    1 DO 2 I=1,JT
      GT(I)=0.0
      GDT(I)=0.0
      SINP(I)=0.0
      COSP(I)=0.0
      ANS(I)=0.0
      PLOG(I)=0.0
      DMOD(I)=0.0
      PHIEQ(I)=0.0
      X(I)=0.0
      Y(I)=0.0
    2 CONTINUE
      DO 3 I=1,25
      PSUM(I)=0.0
      VARY(I)=0.0
      VARX(I)=0.0
    3 CONTINUE
C
C     ESTABLISH CONSTANT COEFFICIENTS AND DELTA TAU STEP SIZE, DTAU.
      PI=3.141592653589793
      PHI=-PI*A
```

```
      DPHI=PI/B
      SIGMA=PI/C
      PHIZRO=PI*PHIZRO
      PHIZRO=PHIZRO*(DSQRT(1.0-(ALPHA*(SIGMA**2))))
      GAMMA=DSIN(PI*GAMMA)
      WRITE(6,193) PHIZRO, GAMMA
  193 FORMAT('0', 'PHIZRO=', D30.16, 5X, 'GAMMA=', D30.16)
      D=(DSQRT(0.5/PI))/SIGMA
      DD=0.5/(SIGMA**2)
      DTAU=(ALPHA/4.0)*DPHI**2
      F=(0.5*DTAU)/DPHI
      NUP=NUM/10
      KK=0
      NA=(N+1)/2
      NAA=NA-1
      NAP1=NA+1
      NM1=N-1
      NM2=N-2
      NM3=N-3
      NM4=N-4
      NM5=N-5
      NB=IDINT(B)
      IB=2*NB
      NBP1=NB+1
      IBP1=IB+1
      J=2
      JJJ=IDINT(A)
      JI=(JJJ+1)/2
      JII=JJJ-1
      IJ=1
C
C     ESTABLISH ARGUMENT VALUES FOR DRAW SUBROUTINE
      ITYPE=0
      IXUP=0
      IYRT=4
      MDXAX=0
      MDYAX=2
      IWIDE=8
      IHIGH=6
      IGRID=1
C
C     LOAD DPHI INCREMENTS INTO PHIEQ(I)
      PHIEQ(NA)=0.0
      DO 190 I=NAP1,N
      NEQ=N+1-I
      II=I-1
      PHIEQ(I)=PHIEQ(II)+DPHI
      PHIEQ(NEQ)=-PHIEQ(I)
```

```
  190 CONTINUE
C
C       LOAD SINE VALUES IN SINP(I) AND COSINE VALUES IN COSP(I)
        DO 12 I=NA,N
        NEQ=N-I+1
        SINP(I)=DSIN(PHIEQ(I))
        SINP(NEQ)=-SINP(I)
        COSP(I)=DCOS(PHIEQ(I))
        COSP(NEQ)=COSP(I)
   12   CONTINUE
C
C       CALCULATE AND LOAD INTO GT(I) INITIAL BOUNDARY CONDITION
        DO 16 I=1,N
        P=-DD*((PHIEQ(I)-PHIZRO)**2)
        GT(I)=D*DEXP(P)
        ANS(I)=GT(I)
   16   CONTINUE
        VARX(1)=ANS(NA)/5.0
   19   AREA=0.0
        DO 31 I=2,NM1
        AREA=AREA+GT(I)
   31   CONTINUE
        AREA=(AREA+(GT(1)+GT(N))/2.0)*DPHI
        TAU=-0.5*DLOG(1.0-ALPHA*(SIGMA**2))
        BLT=TAU/4.0
        MC=0
        NPTS=301
        EXSC=3.0
        YSCL=VARX(1)
        DO 69 I=1,301
        IREV=NA-151+I
        X(I)=PHIEQ(IREV)
        Y(I)=GT(IREV)
   69   CONTINUE
        WRITE(6,22) KK, BLT
   22   FORMAT('1', 'KK=', I6, 5X, 'BLT=', D23.16, 10X,
     $'INITIAL BOUNDARY CONDITION')
        WRITE(6,24)(GT(I),I=1,N)
   24   FORMAT(' ', 4D30.16)
        WRITE(6,33) AREA
   33   FORMAT('0', 'GAUSSIAN AREA=', D23.16)
        WRITE(6,188) TAU
  188   FORMAT('0', 'TAU=', D23.16)
        IIII=0
        GO TO 186
   78   IIII=1
        III=0
        ALPHA=-ALPHA
```

```
      GO TO 90
   79 III=1
      ALPHA=-ALPHA
      DO 81 I=1,N
      GT(I)=ANS(I)
   81 CONTINUE
C
C     GDT=G(PHI,TAU+DTAU) AND GT=G(PHI,TAU)
C     COMPUTE GDT(I) FROM GT(I) USING ALGORITHM, LOAD GDT(I) INTO GT(I)
   25 DO 45 II=1,NUP
      DO 30 I=2,NM1
      IM1=I-1
      IP1=I+1
      GDT(I)=GT(I)+0.25*(GT(IP1)-2.0*GT(I)+GT(IM1))+F*(GAMMA-SINP(I))*
     $(GT(IP1)-GT(IM1))
   30 CONTINUE
C     END-VALUE INTERPOLATION SCHEME
      GDT(1)=(590.*GDT(2)-865.*GDT(3)+660.*GDT(4)-260.*GDT(5)+44.*GDT(6)
     $)/169.0
      GDT(N)=(590.*GDT(NM1)-865.*GDT(NM2)+660.*GDT(NM3)-260.*GDT(NM4)+44
     $.*GDT(NM5))/169.0
   32 DO 35 I=1,N
      GT(I)=GDT(I)
   35 CONTINUE
      KK=KK+1
   45 CONTINUE
      BLT=0.25*(TAU+DTAU*DFLOAT(KK))
      VARX(J)=GT(NA)/5.0
      MC=1
      IF(J.GE.3) MC=2
      IF(KK.GE.NUM) MC=3
      NPTS=701
      EXSC=6.0
      YSCL=VARX(2)
      DO 201 I=1,701
      IREV=NA-351+I
      X(I)=PHIEQ(IREV)
      Y(I)=GT(IREV)
  201 CONTINUE
      WRITE(6,37) KK, BLT
   37 FORMAT('1', 'KK=', I6, 5X, 'BLT=', D23.16, 10X, 'G(PHI,T)')
      WRITE(6,39)(GT(I),I=1,N)
   39 FORMAT(' ', 4D30.16)
      IF(NNN.EQ.1) GO TO 221
      IF(MC.NE.1) GO TO 222
      READ(5,60)(TITLE(I),I=1,6)
      READ(5,60)(TITLE(I),I=7,12)
  222 CONTINUE
```

```
      CALL DRAW(NPTS, X, Y, MC, ITYPE, LABEL, TITLE, EXSC, YSCL, IXUP,
     $IYRT, MDXAX, MDYAX, IWIDE, IHIGH, IGRID, LAST)
  221 CONTINUE
C
C     ANS=P(PHI,TAU)
C     COMPUTE ANS(I)=GT(I)*EXP(ALPHA*(COS(PHI)+GAMMA*PHI))
   90 DO 100 I=1,N
      ABC=ALPHA*(COSP(I)+(GAMMA*PHIEQ(I)))
      ANS(I)=GT(I)*DEXP(ABC)
  100 CONTINUE
      IF(III.EQ.0) GO TO 79
      DO 203 I=1,701
      IREV=NA-351+I
      Y(I)=ANS(IREV)
  203 CONTINUE
      VARY(J)=ANS(NA)/5.0
      YSCL=VARY(2)
      WRITE(6,200) K, BLT
  200 FORMAT('1', 'K=', I6, 5X, 'BLT=', D23.16, 10X, 'P(PHI,T)')
      WRITE(6,300)(ANS(I),I=1,N)
  300 FORMAT(' ', 4D30.16)
      IF(NNN.EQ.0) GO TO 220
      IF(MC.NE.1) GO TO 211
      READ(5,60)(TITLE(I),I=1,6)
      READ(5,60)(TITLE(I),I=7,12)
  211 CONTINUE
      CALL DRAW(NPTS, X, Y, MC, ITYPE, LABEL, TITLE, EXSC, YSCL, IXUP,
     $IYRT, MDXAX, MDYAX, IWIDE, IHIGH, IGRID, LAST)
  220 CONTINUE
C
C     MODULO 2-PI EXPANSION
      DO 83 I=1,IBP1
      DMOD(I)=ANS(I)
      IA=IB
      DO 82 II=1,JII
      IAA=I+IA
      DMOD(I)=DMOD(I)+ANS(IAA)
      IA=IA+IB
   82 CONTINUE
   83 CONTINUE
      VARY(J)=DMOD(NA)/5.0
      DMDMN=0.0
      DMDVAR=0.0
      DO 192 I=1,IBP1
      IREV=NA-NB-1+I
      DMDMN=DMDMN+(DMOD(I)*PHIEQ(IREV))
      DMDVAR=DMDVAR+(DMOD(I)*(PHIEQ(IREV)**2))
  192 CONTINUE
```

```
      DMDMN=DMDMN*DPHI
      DMDVAR=DMDVAR*DPHI
      DMDVAR=DMDVAR-(DMDMN**2)
      PSMARA=0.0
      DO 86 I=2,IB
      PSMARA=PSMARA+DMOD(I)
   86 CONTINUE
      PSMARA=(PSMARA+(DMOD(1)+DMOD(IBP1))/2.0)*DPHI
      WRITE(6,185) KK,BLT
  185 FORMAT('1', 'KK=', I6, 5X, 'BLT=', D23.16, 10X, 'MODULO 2*PI')
      WRITE(6,183)(DMOD(I), I=1,IBP1)
  183 FORMAT(' ', 4D30.16)
      WRITE(6,184) PSMARA
  184 FORMAT('0', 'MODULO 2*PI AREA=', D23.16)
      WRITE(6,194) DMDMN, DMDVAR
  194 FORMAT('0', 'MODULO MEAN=', D23.16, 5X, 'MOD VARIANCE=', D23.16)
      NANA=NA
      IBP1M=(IBP1+1)/2
      DO 44 I=IBP1M,IBP1
      IREV=IBP1+1-I
      X(I)=PHIEQ(NANA)
      X(IREV)=-X(I)
      NANA=NANA+1
   44 CONTINUE
      DO 88 I=1,IBP1
      Y(I)=DMOD(I)
   88 CONTINUE
      EXSC=0.8
      NPTS=IBP1
C
C     SUMMATION ALGORITHM FINDS AREA OF EACH 2*PI INCREMENT OF P(PHI,T)
      NZ=IB+1
      NX=1
      DO 175 I=1,JJJ
      NXP1=NX+1
      NZM1=NZ-1
      PSUM(I)=0.0
      DO 174 II=NXP1,NZM1
      PSUM(I)=PSUM(I)+ANS(II)
  174 CONTINUE
      PSUM(I)=(PSUM(I)+(ANS(NX)+ANS(NZ))/2.0)*DPHI
      NX=NX+IB
      NZ=NZ+IB
  175 CONTINUE
      PSMTOT=0.0
      DO 176 I=1,JJJ
      PSMTOT=PSMTOT+PSUM(I)
  176 CONTINUE
```

```
      WRITE(6,177)
  177 FORMAT('0', 'P(PHI,T) 2*PI INCREMENT AREA VALUES')
      WRITE(6,178)(PSUM(I), I=1,JJJ)
  178 FORMAT(' ', 4D30.16)
      WRITE(6,179) PSMTOT
  179 FORMAT('0', 'PSUM(I) TOTAL=', D23.16)
C
C     MEAN, VARIANCE, AND STANDARD DEVIATION OF P(PHI,T)
  186 PMEAN=0.0
      PDEV=0.0
      DO 180 I=1,N
      PMEAN=PMEAN+(ANS(I)*PHIEQ(I))
      PDEV=PDEV+(ANS(I)*(PHIEQ(I)**2))
  180 CONTINUE
      PMEAN=PMEAN*DPHI
      PDEV=PDEV*DPHI
      PDEV=DSQRT(PDEV-(PMEAN**2))
      PDEVAR=PDEV**2
      WRITE(6,181)
  181 FORMAT('0', 'MEAN, VARIANCE, AND STANDARD DEVIATION OF P(PHI,T)')
      WRITE(6,182) PMEAN, PDEVAR, PDEV
  182 FORMAT('0', 'M(T)=', D23.16, 5X, 'VAR(T)=', D23.16, 5X,
     $'STD DEV(T)=', D23.16)
      IF(IIII.EQ.0) GO TO 78
      IF(KK.GE.NUM) GO TO 50
      J=J+1
      GO TO 25
   50 CONTINUE
      IF(NNN.EQ.1) GO TO 5
   60 FORMAT(6A8)
      STOP
      END
```

BIBLIOGRAPHY

1. Viterbi, A. J., Principles of Coherent Communication, pp. 14-40 and 76-99, McGraw-Hill, 1966.

2. Tikhonov, V. I., "The Effects of Noise on Phase-lock Oscillator Operation," Automat. i Telemakh., v. 22, no. 9, 1959.

3. Ibid., "Phase-lock Automatic Frequency Control Application in the Presence of Noise," Automat i Telemakh., v. 23, no. 3, 1960.

4. La Frieda, J. R., Transient Solutions of the Fokker-Planck Equation For A Class of First-order Phase-Locked Loops, Paper presented at Fourth Hawaii International Conference on System Sciences, Honolulu, Hawaii, pp. 471-473, January 1971. This material also appears on pp. 635-638 in [7].

5. Dominiak, K. E. and Pickholtz, R. L., "Transient Behavior of a Phase-Locked Loop in the Presence of Noise," IEEE Transactions on Communication Technology, v. Com-18, no. 4, pp. 452-456, August 1970.

6. von Neumann, J. and Richtmyer, R. D., "On the Numerical Solution of Partial Differential Equations of Parabolic Type," Los Alamos Report LA657, 25 December 1947. Reprinted in John von Neumann, Collected Works, v. 5, A. H. Taub, Ed., pp. 652-663, Macmillan, 1963.

7. Lindsey, W. C., Synchronization Systems in Communication and Control, pp. 400-410, Prentice-Hall, 1972.

8. Smith, G. D., Numerical Solution of Partial Differential Equations, pp. 70-72, Oxford University Press, 1965.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor John Ohlson, Code 5201 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	3
4.	Ensign Allan Rutherford, USN c/o 2118 Antonio Avenue Camarillo, California 93010	2
5.	Prof. John Geist Code 52GJ Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	LT. Duncan P. Johnson, USCG Box 1749 Naval Postgraduate School Monterey, California 93940	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Naval Postgraduate School Monterey, California 93940	Unclassified
	2b. GROUP

3. REPORT TITLE

Numerical Solution of the Fokker-Planck Equation for The First Order Phase-Locked Loop

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Master's Thesis; September 1972

5. AUTHOR(S) *(First name, middle initial, last name)*

Allan Rutherford

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1972	53	8

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

The transient behavior of the phase error probability density function of a first-order phase-locked loop in the presence of additive white Gaussian noise is determined using numerical techniques. In addition, the modulo-2π probability density function and the cycle slipping density function are found. The algorithm used in these analyses involves numerical integration of a normalized and factored Fokker-Planck equation. Results are shown for cases involving various signal-to-noise ratios and initial conditions.

DD FORM 1 NOV 68 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Phase-Locked Loop						
Fokker-Planck Equation						
Numerical Integration						
Phase-Error Probability Density Function						
Phase Error Modulo-2π Probability Density Function						
Cycle Slipping						

Thesis
R926
c.1

139142

Rutherford
Numerical solution of the Fokker-Planck equation for the first order phase-locked loop.